

Perbandingan Jumlah Pinalti Alokasi Task pada Penjadwalan Kerja dengan Perhitungan Manual dan Algoritma Genetika

Asri Maspupah
Jurusan Informatika, Fakultas MIPA
Universitas Jenderal Achmad Yani
Jl. Terusan Sudirman, Cimahi
asri.maspupah89@gmail.com

Ani Rahmani
Jurusan Teknik Informatika
Politeknik Negeri Bandung
Jl. Gegerkalong Hilir, Ds. Ciwaruga
anirahma@gmail.com

Abstrak—Penelitian ini mengambil studi kasus penjadwalan pengalokasian *task* kepada *resource* pada proyek pendokumentasian alur program di sebuah perusahaan. Alokasi *task* yang dilakukan pada pembuatan jadwal adalah perhitungan alokasi *line of code* harus sesuai dengan target *line of code* yang harus diselesaikan. Permasalahan yang terjadi pada saat pembuatan jadwal adalah perhitungan jumlah *line of code* dari *script program* pada penentuan *task* kadang kala tidak sesuai dengan target *line of code*. Perhitungan yang tidak tepat ini berdampak pada saat pengerjaan dan estimasi total waktu pengerjaan. Oleh karena itu diusulkan pembuatan jadwal dengan menggunakan algoritma genetika. Pendekatan yang dilakukan dengan cara membandingkan apakah pembuatan jadwal dengan algoritma genetika dapat memberikan hasil yang lebih baik dibandingkan dengan proses manual. Algoritma genetika adalah teknik pencarian dan optimasi yang terinspirasi dari prinsip genetika dan seleksi alam yang biasa digunakan untuk memecahkan suatu pencarian nilai pada masalah optimasi. Hasil uji coba penelitian menunjukkan bahwa pembuatan jadwal dengan algoritma genetika ternyata mampu meminimalisir perhitungan jumlah *line of code* yang tidak sesuai dengan target dibandingkan dengan perhitungan manual. Hal ini ditunjukkan dengan adanya penurunan jumlah *line of code* yang tidak sesuai dengan target yaitu sebesar 699 baris atau sebesar 15,9 % dari total. Kontribusi dalam penelitian memberikan kesimpulan bahwa pembuatan jadwal dengan algoritma genetika memberikan hasil yang lebih baik dibandingkan jadwal yang dibuat manual meskipun hasilnya tidak cukup signifikan.

Kata kunci—penjadwalan *task*; algoritma genetika

I. PENDAHULUAN

Penjadwalan adalah alokasi dari sumber daya (*resource*) terhadap waktu untuk menghasilkan sebuah kumpulan pekerjaan (*task*) (Kenneth, 1994). Obyek dari penelitian ini adalah mengambil sebuah studi kasus yaitu proyek penjadwalan kerja mengenai pembuatan *program flow* dari sebuah *script program* di suatu perusahaan. Tujuan dari proyek tersebut yaitu melakukan *reverse engineering* dengan melakukan pendokumentasian *program flow* dari kode program yang sudah ada. Penentuan jadwal kerja pada proyek tersebut berdasarkan pada perhitungan jumlah *line of code* (baris kode program) atau disingkat LOC dari *script program* yang harus sesuai dengan target jumlah LOC perharinya. Pada penelitian ini, istilah penjadwalan kerja tersebut dikenakan istilah *time scheduler script program*.

Time scheduler script program adalah suatu proses pengalokasian *resource* untuk menjalankan sekumpulan pekerjaan (*task*) dalam jangka waktu tertentu sesuai dengan target yang harus di selesaikan oleh *resource* dalam waktu satu hari. Dalam membuat jadwal ada beberapa parameter yang dibutuhkan [2] yaitu;

- Berapa banyak *resource* yang terlibat dalam pembuatan *script program*.
- Berapa banyak *script program* yang harus di selesaikan dalam jadwal kerja.
- Jumlah LOC yang harus dikerjakan oleh satu PIC dalam satu hari. Penentuan target LOC tergantung pada kebijakan manager. Semakin banyak LOC yang harus dikerjakan maka semakin cepat pula pengerjaan

proyeknya tetapi harus berbanding lurus terhadap kemampuan masing-masing PIC. Oleh karena itu, seorang manager harus mempertimbangkan dengan benar dalam menentukan parameter ini.

d. Waktu pengerjaan task. Waktu pengerjaan task sesuai dengan jam kerja kantor.

Permasalahan pada *time scheduler script program* ini cukup kompleks disebabkan oleh variasi jumlah LOC dari setiap *script program*. Variasi dari LOC tersebut dapat menyebabkan perhitungan LOC pada jadwal tidak sesuai dengan jumlah LOC yang ditargetkan dalam satu hari. Perhitungan alokasi LOC yang tidak tepat ini akan berdampak pada saat proses pengerjaan dan estimasi waktu pengerjaan proyek. Jika alokasi LOC melebihi target, berdampak pada usaha pengerjaan yaitu penambahan waktu pengerjaan di luar jam kerja. Sedangkan jika alokasi LOC yang tidak mencapai target, berdampak pada *scheduler* yakni estimasi waktu selesainya proyek menjadi lebih lama. Contoh perhitungan LOC pada jadwal ditunjukkan pada Tabel 1.

Dari analisa perhitungan LOC pada Tabel 1 terdapat dua kondisi alokasi LOC yang dikatakan tidak sesuai dengan target. Kondisi tersebut adalah kondisi alokasi LOC yang kurang dari target LOC per hari dan kondisi alokasi LOC yang melebihi dari target LOC per hari. Selisih angka ini cukup signifikan karena akan mempengaruhi kecepatan pengerjaan. Lebih lanjut lagi mengenai istilah “pengalokasian LOC yang tidak tepat” dikenakan istilah “pinalti” sebagai pendekatan yang digunakan dalam penelitian ini.

TABEL 1. PERHITUNGAN LOC PADA JADWAL DENGAN MANUAL

| Tanggal | PIC | Index Task | List Task | Plan LOC / hari | Total LOC | JML LOC Tidak Tepat | Target |
|-------------|-------|------------|-----------|-----------------|-----------|---------------------|--------|
| | | | | | | | 1000 |
| 23 Feb 2012 | PIC 1 | 380 | FMC00004 | 1057 | 1057 | 57 | |
| 23 Feb 2012 | PIC 2 | 395 | FOD00001 | 95 | 1500 | 500 | |
| | PIC 2 | 396 | FOD00002 | 30 | | | |
| | PIC 2 | 397 | FOD00003 | 96 | | | |
| | PIC 2 | 398 | FOD00004 | 30 | | | |
| | PIC 2 | 400 | FHR00004 | 844 | | | |
| | PIC 2 | 401 | FOD00005 | 96 | | | |
| | PIC 2 | 402 | FOD00006 | 30 | | | |
| | PIC 2 | 403 | FOD00007 | 30 | | | |
| | PIC 2 | 399 | FJQ00003 | 249 | | | |
| 23 Feb 2012 | PIC 3 | 377 | FGD00014 | 313 | 1090 | 90 | |
| | | 378 | FIM00002 | 777 | | | |
| 23 Feb 2012 | PIC 4 | 387 | FME00043 | 1000 | 1000 | 0 | |
| 23 Feb 2012 | PIC 5 | 379 | FFD00084 | 162 | 1000 | 0 | |
| | | 383 | FHI00005 | 643 | | | |
| | | 384 | FOC00001 | 195 | | | |
| 07 Mar 2012 | PIC 1 | 404 | FOC00002 | 607 | 1023 | 23 | |
| | PIC 1 | 405 | FOC00003 | 59 | | | |
| | PIC 1 | 406 | FOD00008 | 357 | | | |
| 07 Mar 2012 | PIC 2 | 446 | FME00044 | 330 | 755 | -245 | |
| | PIC 2 | 447 | FMC00011 | 425 | | | |
| 07 Mar 2012 | PIC 3 | 385 | FIR00008 | 1000 | 1000 | 0 | |
| 07 Mar 2012 | PIC 4 | 387 | FME00043 | 1000 | 1000 | 0 | |
| 07 Mar 2012 | PIC 5 | 411 | FNK00022 | 493 | 1004 | 4 | |
| | PIC 5 | 412 | FOD00010 | 320 | | | |
| | PIC 5 | 413 | FHR00007 | 191 | | | |

Pada umumnya solusi masalah *time scheduler script program* dapat diperoleh dengan metode heuristik, diantaranya menggunakan Metode Rank Based Ant System (ASrank) dan Algoritma Genetika. Penelitian ini menggunakan algoritma genetika untuk mencari solusi pada permasalahan di atas. Algoritma genetika adalah metode heuristik berdasarkan pada mekanisme seleksi alam dan proses evolusi alam [1]. Beberapa penelitian sebelumnya menerapkan algoritma genetika dalam pembuatan jadwal kuliah [2], kombinasi lahan tanah untuk properti [3] dan optimalisasi kandungan karbohidrat dan protein pada mocorin [4].

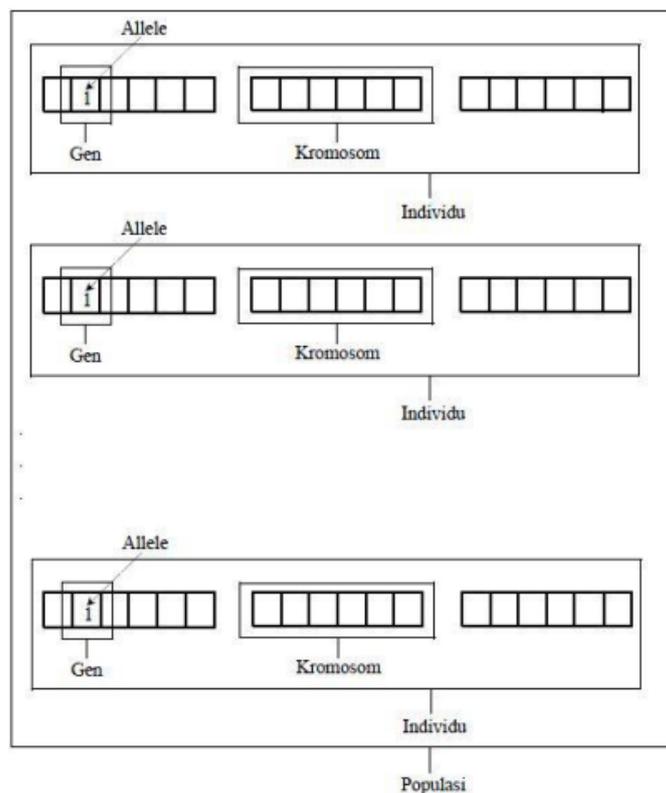
Penelitian pada *time scheduler script program* dengan menggunakan algoritma genetika bertujuan untuk meminimalisir pengalokasian task yang jumlah LOC-nya tidak sesuai dengan jumlah LOC yang ditargetkan dalam waktu satu hari. Selain itu, tujuan penelitian lain adalah untuk menguji hipotesa bahwa apakah benar dengan menggunakan algoritma genetika dapat mengurangi jumlah alokasi LOC yang tidak sesuai dengan target pada jadwal yang masih menggunakan perhitungan manual.

Makalah utama terdiri dari empat bagian : I. Pendahuluan. II. Metode. III. Hasil dan Diskusi, IV. Kesimpulan. Bagian berikutnya adalah Daftar Pustaka. Bagian pendahuluan menjelaskan latar belakang dan tujuan dari penelitian. Bagian Metode menunjukkan metode dari penelitian yang dilakukan. Bagian Hasil dan Diskusi menunjukkan hasil dari penelitian. Terakhir, bagian kesimpulan menjelaskan inti sari keseluruhan penelitian.

II. METODE

Ide dasar dari algoritma genetika adalah proses evolusi yang terinspirasi dari teori evolusi Darwin yang menyatakan kelangsungan hidup satu makhluk dipengaruhi aturan bahwa yang kuat adalah yang menang. Algoritma genetika mengikuti prosedur/tahap-tahap yang menyerupai proses evolusi, yaitu adanya proses seleksi, crossover dan mutasi. Algoritma genetika sangat tepat digunakan untuk penyelesaian masalah optimasi yang kompleks dan sulit diselesaikan dengan metode konvensional. Algoritma genetika yang digunakan untuk menyelesaikan permasalahan pada obyek penelitian ini adalah Simple Genetic Algorithm (SGA).

Representasi data yang digunakan pada algoritma genetika untuk masalah *time scheduler program* adalah representasi data terstruktur. Data disajikan dalam bentuk rangkaian barisan data yang disebut dengan populasi [5]. Sedangkan dalam satu rangkaian merepresentasikan individu yang dikenal dengan sebutan kromosom. Kromosom terdiri dari kumpulan gen yang terdiri dari sekumpulan nilai. Nilai dari gen disebut sebagai Allele. Allele sebagai nilai Gen dalam Kromosom merepresentasikan *index task*, nama *script program*, dan jumlah nilai LOC dari *script program*. Dengan demikian, kromosom tersebut merepresentasikan alokasi LOC yang harus dikerjakan oleh *resource* dalam waktu satu hari. Ilustrasi Algoritma Genetika ditunjukkan pada Gambar 1. Penerapan algoritma genetika pada penelitian ditunjukkan pada Tabel 2 dan Gambar 2.



Gambar 1. Ilustrasi Algoritma Genetika [6]

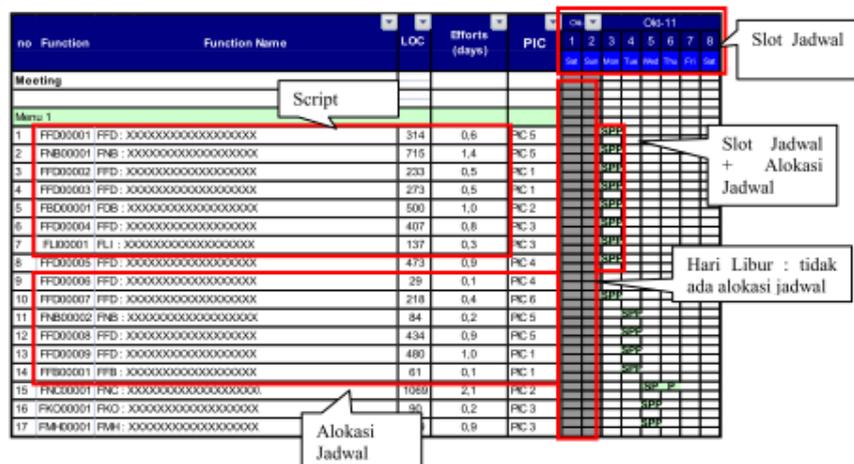
Penerapan Algoritma Genetika dalam penelitian dilakukan dengan dua cara yaitu pertama pemetaan istilah Algoritma Genetika dan pemetaan operasi kromosomnya. Pertama, Pemetaan istilah Algoritma Genetika artinya memetakan karakteristik *time scheduler script program* terhadap istilah-istilah yang ada pada Algoritma Genetika. Pemetaan istilah ini ditunjukkan pada Tabel 2. Kemudian pemetaan dilanjutkan dengan representasi kromosom

sesuai dengan lingkup obyek penelitian. Pemetaan kromosom harus dapat memberikan informasi yang dibutuhkan untuk konstruksi seluruh *time scheduler script program*. Representasi kromosom ditunjukkan pada Gambar 2.

Gambar 2 memperlihatkan bahwa setiap script dipetakan terhadap PIC yang *available* yaitu belum memiliki jadwal. Oleh karena itu, pada pemetaan jadwal terhadap kromosom memiliki 3 dimensi yaitu PIC, Waktu dan Script. Keterangan Slot Jadwal dan Slot Jadwal + Alokasi Jadwal pada Gambar 2 ditunjukkan pada Gambar 3.

TABEL 2. PEMETAAN OBYEK PENELITIAN KE DALAM ISTILAH ALGORITMA GENETIKA

| Istilah dalam algoritma genetika | Istilah dalam penjadwalan | |
|----------------------------------|------------------------------|--|
| Gen | Tanggal, Script Program, PIC | |
| Allele | Tanggal | 1 October 2011, 2 October 2011dst |
| | Script (Nama Script, LOC) | (ABC001, 506), (ABD003, 102) dst |
| | PIC | PIC1, PIC2, dst |
| Kromosom | Slot jadwal | Gen Tanggal |
| | Alokasi jadwal | Gen Script dan PIC |
| Individu | Jadwal | Alokasi jadwal yang sudah ditempatkan pada slot jadwal |
| Populasi | List Jadwal | Kumpulan jadwal yang akan menjadi solusi |



Gambar 2. Representasi Kromosom *Time Sceduler Script Program*

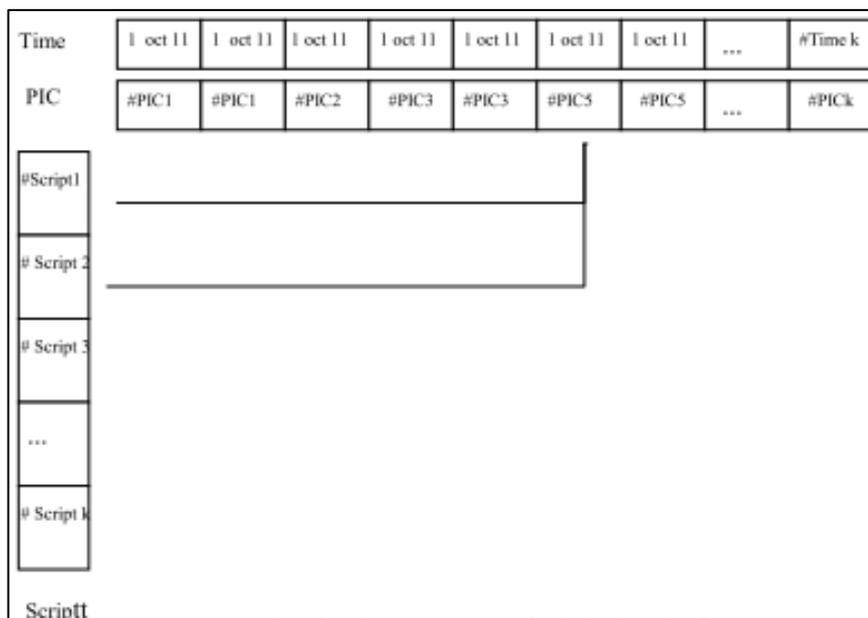
| Slot jadwal | | | | Slot Jadwal + Alokasi Jadwal | | | | |
|-------------|---------|-------|---------|------------------------------|-------------|-----|------------|-----|
| Hari | Tanggal | Bulan | Tahun | Script | Description | LOC | Effort Day | PIC |
| Saturday | | 1 | Oktober | 2011 | | | | |
| Sunday | | 2 | Oktober | 2011 | | | | |
| Monday | | 3 | Oktober | 2011 | | | | |
| Start time | | | | | | | | |
| Hari | Tanggal | Bulan | Tahun | | | | | |
| Monday | | 3 | Oktober | 2011 | | | | |
| Finish Time | | | | | | | | |
| Hari | Tanggal | Bulan | Tahun | | | | | |
| Monday | | 3 | Oktober | 2011 | | | | |

Gambar 3. Representasi Slot Jadwal

Kedua, Pemetaan operasi kromosom sebagai operasi genetika pada penelitian ini dibagi menjadi dua tahap. Pertama, tahap pemerataan pengalokasian *script program* terhadap semua PIC. Pada tahap ini, *script program* yang sudah dialokasi belum dijadwalkan terhadap target LOC untuk setiap harinya. Kedua, pembuatan jadwal harian dari sekumpulan *script program* yang sudah di alokasikan pada setiap PIC (*Ouput* dari Tahap pertama). Dengan demikian pada kedua tahap tersebut akan dilakukan operasi genetika.

Pemetaan operasi kromosom pada tahap pertama, pengalokasian *script program* terhadap semua PIC, adalah kegiatan dalam rangka pemerataan *script program* yang ada untuk dialokasikan ke setiap PIC berdasarkan jumlah LOC. Dengan asumsi bahwa setiap PIC mempunyai level yang sama sehingga pengalokasian harus dibagi sama rata. Pada tahap ini menggunakan SGA yang berulang pada n iterasi PIC. Hal ini berarti bahwa alokasi *script* dilakukan dari PIC 1 (pertama) sampai PIC n (last PIC). Output yang dihasilkan dari kegiatan ini adalah keterhubungan antara *script* dan PIC yang belum dialokasikan terhadap slot jadwal.

Pemetaan operasi kromosom pada tahap kedua, pembentukan slot jadwal dari alokasi program pada semua PIC, adalah kegiatan dalam rangka pembuatan jadwal harian dari sekumpulan *script program* yang sudah dialokasikan pada tiap PIC (*ouput* dari tahap pertama). Pembuatan slot jadwal ini bergantung pada target LOC yang harus dikerjakan oleh setiap PIC. Pada tahap ini menggunakan SGA yang berulang pada n iterasi *script program* yang ada pada masing – masing PIC, artinya penentuan slot jadwal dilakukan dari Script 1 (pertama) sampai Script n (terakhir). Output yang dihasilkan dalam kegiatan ini adalah keterhubungan antara alokasi jadwal (*script* dan PIC) dengan waktu sehingga terbentuk slot jadwal. Representasi alokasi jadwal ditunjukkan pada Gambar 4.



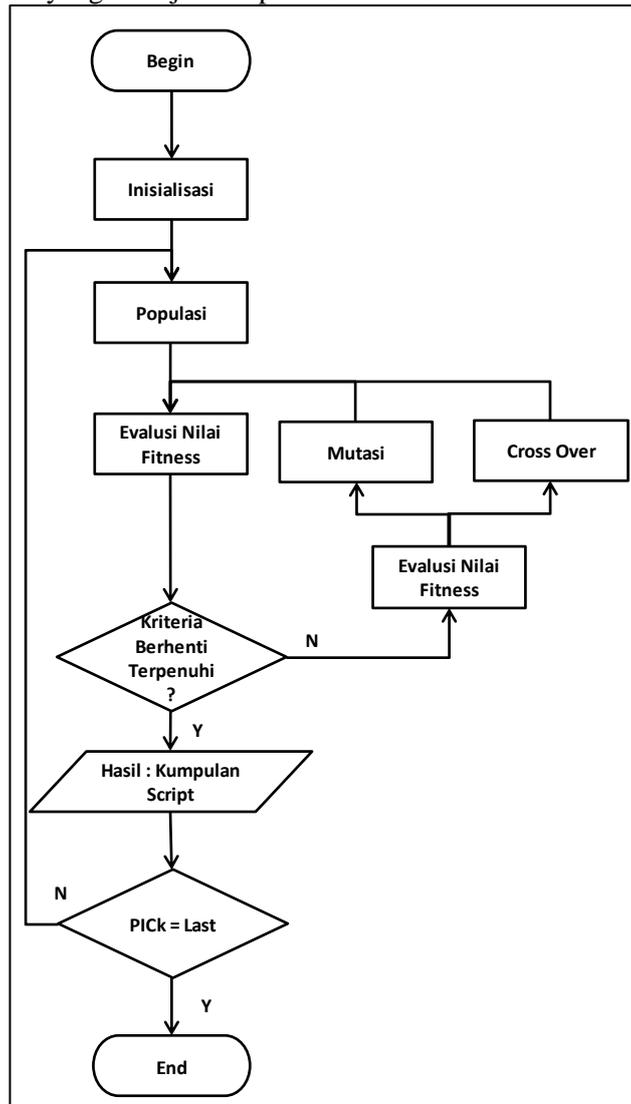
Gambar 4. Representasi Alokasi Jadwal

Pada saat menjalankan operasi genetika dibutuhkan parameter algoritma genetika. Parameter algoritma genetika digunakan untuk mengontrol proses genetika dalam pertukaran gen atau penyebaran kromosom, seperti ukuran populasi ($popsiz$), probabilitas crossover (peluang crossover - ρ_c) dan probabilitas mutasi (peluang mutasi - ρ_m). Proses genetika yang dilakukan pada penelitian ini berdasarkan pada *fitness* dari individu terbaik yang ditinjau pada setiap generasi [7]. Oleh karena itu, parameter yang digunakan adalah: ($popsiz$; ρ_c ; ρ_m) = (80; 0,45; 0,01).

SGA dimulai dengan membangkitkan populasi dari sekumpulan kromosom sebagai sebagai solusi awal pada penjadwalan [6]. Populasi awal dibentuk menggunakan teknik *random value* dengan melakukan pemilihan acak terhadap sekumpulan *script program* [4]. Jumlah kromosom populasi awal pada tahap pertama dibatasi dengan sejumlah *script program* yang harus dikerjakan oleh seorang *resource* yaitu total jumlah seluruh *script program* dibagi dengan banyaknya *resource*. Sedangkan jumlah kromosom pada tahap kedua yakni antara 1 s/d 5 buah. Proses selanjutnya analog pada proses evolusi alam yaitu seleksi, elisitisme, crossover dan mutasi.

Proses seleksi, crossover dan mutasi terus berlangsung sampai menemukan solusi mendekati optimum dengan nilai pinalti sama dengan 0 (nol) atau jika didapatkan nilai *fitness* yang konvergen ke suatu nilai secara berturut –

berturut [8]. Secara umum tahap-tahap algoritma genetika untuk penyelesaian masalah *time scheduler script program* terlihat pada *flow chart* yang ditunjukkan pada Gambar 5.



Gambar 5. Flow Chart Algoritma Genetika untuk Penjadwalan Alokasi Task

Penjelasan detail mengenai seleksi, elitisme, crossover dan mutasi dijelaskan dibawah ini;

A. Seleksi

Seleksi adalah proses pemilihan individu terbaik dari sekumpulan populasi [8] [9]. Kriteria yang digunakan dalam proses seleksi ini adalah nilai *fitness*. Setiap individu dilakukan proses penghitungan total alokasi LOC dengan menjumlahkan seluruh gen LOC pada kromosom. Kemudian dari perhitungan ini akan dihitung nilai pinalti dan *fitness*-nya. Tahapan perhitungan *fitness* adalah sebagai berikut:

Pada tahap pertama;

1. Menghitung target LOC ($nSPIC$) = $nLS / nPIC$

Ket: $nSPIC$: Jumlah LOC dari *script program* yang harus di kerjakan oleh setiap PIC

nLS : Jumlah seluruh LOC dari *script program*

$nPIC$: Jumlah *resource* atau PIC

2. Mencari total LOC $LOC_i = \sum loc_i$

3. Mencari pinalti = $|nSPIC - loc|$

4. Mencari nilai *fitness*,

$$Fitness = \frac{1}{1+Penalti}$$

Pada tahap kedua;

1. Menghitung target LOC harian = 1000 (sudah diinisialisasi diawal)
2. Menghitung remaining loc tidak habis dibagi target LOC harian yaitu $LOC \text{ mod } nLocHari$
3. Mencari total LOC $\approx \sum remainingLoc_i$
4. Mencari pinalti = $|\frac{1}{n} \sum remainingLoc_i - (loc \text{ mod } nLocHari)|$
5. Mencari nilai fitness,

$$Fitness = \frac{1}{1+Penalti}$$

Jika ternyata dari proses seleksi tersebut sudah mencapai nilai optimum, dengan nilai pinalti 0 (nol), maka proses genetika berhenti [7]. Namun jika ternyata belum, maka dilakukan proses pembuatan generasi baru.

Contoh perhitungan nilai *fitness*,

Diketahui populasi solusi penjadwalan pada tabel 3 dan LOC yang ditargetkan kepada setiap PIC adalah 3180. Maka berapa nilai *fitness* pada masing – masing individu dalam populasi tersebut ?

TABEL 3. POPULASI PENJADWALAN

| No Individu | List Script | | | | | | | | Total LOC |
|-------------|-------------|------|----|------|----|------|----|------|-----------|
| | S1 | | S2 | | S3 | | S4 | | |
| 1 | a | 100 | b | 1200 | C | 500 | d | 300 | 2100 |
| 2 | b | 1200 | e | 2000 | d | 300 | f | 1500 | 5000 |
| 3 | a | 100 | g | 900 | b | 1200 | h | 1000 | 3200 |
| 4 | k | 500 | e | 2000 | J | 100 | l | 300 | 2900 |
| 5 | a | 100 | b | 1200 | E | 2000 | d | 300 | 4300 |

Untuk menghitung nilai *fitness*, pertama jumlahkan LOC dari setiap kombinasi *script program* pada individu. Ambil contoh dari Tabel 3 no 1 yang memiliki kombinasi script program a,b,c dan d, didapatkan total LOC adalah 2100 (100+1200+500+ 300). Kemudian didapatkan nilai pinaltinya 1080 dengan mencari selisih antara target LOC per PIC dengan total LOC pada individu ke 1 (3180 -2100). Terakhir didapatkan nilai *fitness* 0,000925 dengan menggunakan persamaan 5 ($1 / (1 + pinalti)$). Contoh hasil evaluasi fitness pada Tabel 3 ditunjukkan pada Tabel 4.

TABEL 4. HASIL EVALUASI *FITNESS*

| No Individu | Selisi LOC | Nilai fitness |
|-------------|------------|---------------|
| 1 | 1080 | 0,000925 |
| 2 | 1820 | 0,000549 |
| 3 | 20 | 0,0476 |
| 4 | 280 | 0,003559 |
| 5 | 1120 | 0,000892 |

B. Elitisme

Elitisme adalah proses mempertahankan individu (kombinasi *script program*) terbaik dari setiap generasi [6]. Individu terbaik adalah individu yang memiliki nilai *fitness* tertinggi atau individu yang memiliki nilai pinalti terkecil dari individu lainnya dalam satu populasi. Contoh penggunaan elisitisme pada populasi penjadwalan (Tabel 3).

Dari populasi pada tabel 3 diatas terdapat 5 individu dengan masing – masing nilai fitness-nya yaitu 0,000925, 0,000549, 0,0476, 0,003559 dan 0,000892. Berdasarkan data tersebut didapatkan nilai fitness yang tertinggi pada individu ke tiga yang fitness-nya 0,0476 dengan jumlah pelanggarannya sebanyak 20. Dengan demikian individu ke 3 akan dipertahankan pada pembentukan populasi generasi selanjutnya.

C. Crossover (Penyilangan)

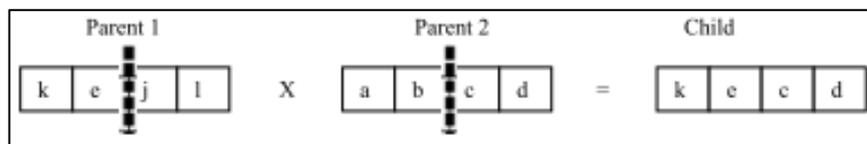
Crossover adalah proses menggabungkan dua kromosom orang tua (*parent*) menjadi kromosom baru (*offspring*) [9]. Dengan demikian teknik crossover akan menukarkan satu atau lebih *script program* pada individu

terpilih pada suatu populasi. Individu yang sudah terpilih tersebut akan dilakukan crossover jika memenuhi parameter genetika. Parameter genetika berfungsi sebagai pengontrol apakah individu tersebut berhak melakukan pertukaran kromosom atau tidak. Parameter genetika tersebut disebut dengan probabilitas crossover (peluang crossover - pc). Metode crossover yang digunakan dalam pertukaran kromosom adalah dengan menggunakan one point crossover.

Proses pemilihan calon *parent* dilakukan dengan menggunakan seleksi *tournament*. Turnamen adalah proses pemilihan kelompok individu sebanyak nilai *tour*. Kemudian dipilih satu individu terbaik dengan *fitness* tertinggi dari kelompok calon *parent* tersebut. Kemudian ditentukan probabilitas *crossover* dengan bilangan random. Jika hasil probabilitas *crossover* lebih kecil dibandingkan dengan probabilitas *crossover* pada parameter operator genetika maka dilakukan *crossover*. Metode *crossover* yang digunakan dalam penelitian ini yakni *one point crossover* yang ditunjukkan pada Gambar 6.

Contoh diberikan *parent* dengan urutan kromosom $I1 = (k,500|e,2000|j,100|l,300)$ dan $I2 = (a,100|b,1200|c,500|d,300)$ serta *cut point* pada posisi 2. *Offspring*-nya dihasilkan dengan cara sebagai berikut:

1. Susunan *script program* pada kromosom I1 disalin ke *offspring* dari posisi awal (1) sampai dengan posisi *cut point* (2), dihasilkan $O = (k,500|e,2000|X|X)$
2. Susunan *script program* pada kromosom I2 disalin ke *offspring* dari posisi *cut point* (2) sampai dengan posisi terakhir (4), dihasilkan $O = (k,500|e,2000|c,500|d,300)$



Gambar 6. Mekanisme Metode *Crossover*

D. Mutasi

Mutasi adalah metode tahap penyebaran kromosom dengan mengganti kromosom yang hilang akibat *crossover*. Dengan kata lain teknik mutasi akan mengganti satu atau lebih *script program* pada seluruh individu suatu populasi. Kromosom yang ada pada individu tersebut akan dilakukan mutasi jika memenuhi parameter genetika. Parameter genetika berfungsi sebagai pengontrol apakah kromosom tersebut berhak dilakukan penggantian atau tidak. Parameter genetika tersebut disebut dengan probabilitas mutasi (peluang mutasi - pm). Metode mutasi yang digunakan adalah pada tingkat kromosom yaitu semua gen dalam kromosom tersebut memiliki kemungkinan untuk diganti dengan gen lain berdasarkan nilai probabilitas mutasi yang didapat dari bilangan random. Jika nilai probabilitas mutasinya lebih kecil dibandingkan dengan probabilitas mutasi pada operator genetika maka proses mutasi dilakukan. Mekanisme metode mutasi ditunjukkan pada Gambar 7.

Contoh diberikan sebuah kromosom $I = (k,500|e,2000|c,500|d,300)$, probabilitas mutasi pada masing-masing gen dalam kromosom tersebut adalah $(0,22|0,24|0,000246|0,53)$. Ternyata yang dilakukan mutasi adalah gen 3 dan akan diganti dengan gen $l,300$. Sehingga kromosom baru nya menjadi $(k,500|e,2000|l,300|d,300)$.

Pemilihan bilangan acak untuk probabilitas pada setiap kromosom

| No Kromosom | Pm | $Pm < pm$ (0,01) | Kromosom Pengganti |
|-------------|----------|------------------|--------------------|
| 1 | 0,22 | Tidak | |
| 2 | 0,24 | Tidak | |
| 3 | 0,000246 | Benar | l |
| 4 | 0,53 | Tidak | |



Gambar 7. Mekanisme Metode Mutasi

E. Evaluasi titik berhenti

Evaluasi titik berhenti adalah tahap evaluasi nilai fitness dari setiap kromosom dalam suatu generasi untuk mencapai titik berhenti. Evaluasi ini dilakukan dengan memeriksa nilai fitness yang paling tinggi pada suatu populasi apakah sudah mencapai solusi optimum. Solusi optimum adalah suatu solusi yang tidak memiliki pelanggaran (pinalti).

Berdasarkan rumus fitness pada pembahasan sebelumnya maka individu yang memiliki fitness dengan nilai 1 dianggap solusi optimum. Angka ini didapat dengan memasukan pinalti = 0 ke dalam persamaan

$$Fitness = \frac{1}{1+Penalti}$$

Perhitungan persamaan :

$$Pinalti = 0;$$

$$Fitness = 1 / (1 + pinalti) = 1 / (1 + 0) = 1 / (1) = 1$$

Dengan demikian selama nilai fitness tertinggi pada suatu populasi lebih kecil dari 1, akan dilakukan terus pembuatan generasi baru. Tetapi timbul masalah lain, bagaimana jika nilai fitness tersebut tidak mencapai solusi optimum tentu saja akan terjadi pembuatan generasi baru yang tidak pernah berujung. Dari sebuah percobaan menunjukkan bahwa jika nilai fitness tidak mencapai solusi optimum, akan konvergen ke suatu nilai. Dengan demikian kriteria berhenti pada algoritma genetika yang digunakan pada penelitian ini adalah,

1. Berhenti bila dalam n generasi berikut mencapai nilai fitness optimum.
2. Berhenti setelah dalam beberapa generasi berturut - turut didapatkan nilai fitness tertinggi tidak berubah (konvergen). Pada penelitian ini digunakan nilai fitness yang konvergen sebanyak 10 kali untuk menentukan kriteria berhenti, yakni selama nilai fitness yang konvergen kurang dari 10 kali, akan terus dilakukan

III. HASIL DAN DISKUSI

Untuk lebih memahami pemakaian algoritma genetika dalam mencari solusi masalah *time scheduler script program*, dilakukan penelitian dengan menggunakan data *sample* penelitian. Data yang digunakan adalah jadwal kerja *creation product documentation* pada sebuah proyek di sebuah perusahaan selama 5 bulan yakni terhitung dari bulan November 2011 sampai dengan bulan Maret 2012. Informasi data penelitian tersebut adalah,

1. Jumlah resource (PIC) : 5,
2. Jumlah task : 454 buah script program dengan 550.163 LOC,
3. Jumlah target harian : 1000 LOC,

Penelitian dilakukan dengan mengamati data jumlah pinalti pada *time shceduler script program* yang dibuat dengan proses manual dan dengan menggunakan algoritma genetika. Tahap selanjutnya adalah membandingkan kedua data tersebut berdasarkan parameter tepat sasaran dan jumlah pinalti.

Definisi Pinalti yang dimaksud pada penelitian ini adalah jumlah LOC yang tidak sesuai dengan target harian. Kondisi pinalti pada jadwal dibagi menjadi dua yakni pinalti pada alokasi LOC tidak mencapai target dan pinalti pada alokasi LOC melebihi target. Contoh kondisi pinalti ditunjukkan pada Tabel 5. Nilai 6 pada kolom Pinalti menunjukkan LOC yang dialokasikan melebihi dari target seharusnya. Sedangkan nilai pinalti -3 menunjukkan LOC yang dialokasikan kurang dari target seharusnya. Pinalti dengan nilai 0 menunjukkan bahwa alokasi LOC dengan target sudah sesuai.

TABEL 5. ALOKASI LOC PADA JADWAL

| Tanggal | PIC | Alokasi LOC (Hari) | Pinalti |
|-------------|-------|--------------------|---------|
| 01 nov 2011 | PIC 1 | 1006 | 6 |
| 01 nov 2011 | PIC 2 | 997 | -3 |
| 01 nov 2011 | PIC 3 | 1000 | 0 |

Proses perhitungan pinalti pada jadwal dilakukan dengan menganalisa alokasi penyebaran *script program* yang akan di kerjakan oleh setiap PIC. Analisa dilakukan dengan menghitung kelebihan dan kekurangan LOC harian

dari hasil pembuatan *time scheduler script program* yang dibuat dengan proses manual maupun dengan menggunakan algoritma Genetika.

A. Hasil perhitungan jumlah pinalti pada jadwal yang dibuat melalui perhitungan manual

Data Tabel 6 memberikan informasi bahwa;

- a. Total jadwal: Alokasi *script* program + PIC + Waktu → 549 buah
- b. Total Jadwal yang alokasi LOC-nya sama persis dengan target → 412 buah dengan persentasi dari total jadwal sebanyak 75,05 %
- c. Total Jadwal yang alokasi LOC-nya melebihi target → 29 buah dengan persentasi dari total jadwal sebanyak 5,28 %
- d. Total Jadwal yang alokasi LOC-nya tidak mencapai target → 108 buah dengan persentasi dari total jadwal sebanyak 19,67 %

TABEL 6. DATA PINALTI PADA JADWAL MANUAL

| Pinalti (-) | Jumlah | Pinalti (+) | Jumlah | Pinalti (+) | Jumlah |
|-------------|--------|-------------|--------|-------------|--------|
| -441 | 1 | 0 | 412 | 26 | 1 |
| -271 | 1 | 1 | 7 | 28 | 2 |
| -245 | 1 | 2 | 3 | 29 | 1 |
| -157 | 1 | 3 | 13 | 30 | 1 |
| -135 | 1 | 4 | 7 | 31 | 1 |
| -79 | 1 | 5 | 4 | 33 | 1 |
| -64 | 1 | 6 | 6 | 38 | 2 |
| -58 | 1 | 7 | 7 | 40 | 1 |
| -53 | 1 | 8 | 4 | 42 | 1 |
| -51 | 1 | 9 | 2 | 43 | 1 |
| -27 | 1 | 10 | 6 | 48 | 1 |
| -23 | 1 | 11 | 3 | 54 | 1 |
| -14 | 1 | 13 | 3 | 55 | 1 |
| -11 | 1 | 14 | 3 | 57 | 1 |
| -9 | 1 | 15 | 1 | 67 | 1 |
| -5 | 1 | 16 | 3 | 82 | 1 |
| -4 | 2 | 17 | 1 | 90 | 1 |
| -3 | 5 | 18 | 6 | 119 | 1 |
| -1 | 6 | 20 | 3 | 499 | 1 |
| | | 21 | 2 | 500 | 1 |
| | | 23 | 2 | | |

B. Hasil perhitungan jumlah pinalti pada jadwal yang dibuat dengan menggunakan algoritma genetika

TABEL 7. DATA PINALTI PADA JADWAL DENGAN MENGGUNAKAN ALGORITMA GENETIKA

| Pinalti (-) | Jumlah | Pinalti (-) | Jumlah | Pinalti (+) | Jumlah | Pinalti (+) | Jumlah |
|-------------|--------|-------------|--------|-------------|--------|-------------|--------|
| -328 | 1 | -21 | 1 | 0 | 382 | 18 | 3 |
| -101 | 1 | -20 | 1 | 1 | 10 | 21 | 1 |
| -97 | 1 | -19 | 4 | 2 | 9 | 22 | 2 |
| -75 | 1 | -18 | 1 | 3 | 6 | 24 | 1 |
| -71 | 2 | -16 | 2 | 4 | 5 | 28 | 1 |
| -62 | 1 | -15 | 2 | 4 | 5 | 30 | 2 |
| -61 | 1 | -13 | 3 | 5 | 3 | 36 | 1 |
| -48 | 1 | -12 | 3 | 6 | 6 | 39 | 1 |
| -46 | 1 | -10 | 1 | 7 | 3 | 40 | 1 |
| -45 | 1 | -9 | 5 | 9 | 3 | 44 | 1 |
| -38 | 1 | -8 | 5 | 10 | 3 | 61 | 1 |
| -33 | 2 | -7 | 3 | 11 | 1 | 76 | 1 |
| -32 | 1 | -6 | 4 | 12 | 1 | 136 | 1 |
| -30 | 1 | -5 | 3 | 13 | 1 | 175 | 1 |
| -28 | 1 | -4 | 5 | 14 | 2 | 207 | 1 |
| -24 | 1 | -3 | 7 | 15 | 1 | 253 | 1 |
| -23 | 1 | -2 | 9 | 16 | 2 | 311 | 1 |
| -22 | 1 | -1 | 11 | 17 | 1 | | |

Data Tabel 7 memberikan informasi bahwa;

- a) Total jadwal: Alokasi script program + PIC + Waktu → 550 buah
- b) Total Jadwal yang tepat → 379 buah dengan persentasi dari total jadwal sebanyak 68.91 %
- c) Total jadwal yang tidak tepat → 171 buah dengan persentasi dari total jadwal sebanyak 31,09 %

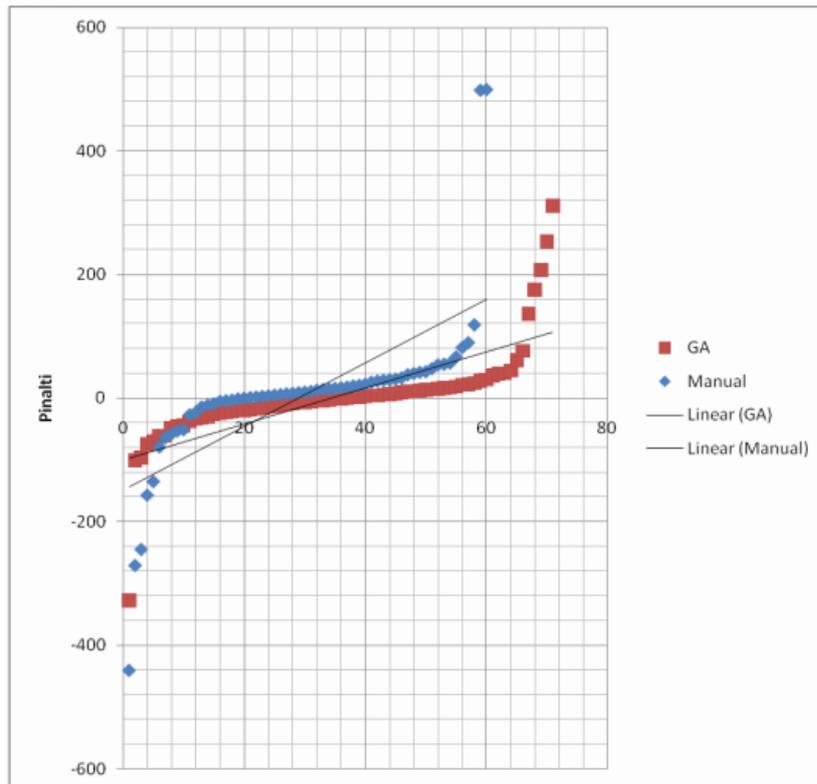
C. Perbandingan jadwal

Tahap perbandingan jadwal adalah tahap evaluasi untuk mengetahui solusi optimum dalam pembuatan penjadwalan dengan membandingkan antara jadwal yang dibuat manual dengan jadwal yang dibuat dengan bantuan SGA (Simple Genetic Algorithm). Perbandingan jadwal dilakukan dengan melakukan perbandingan total jadwal yang tepat sasaran dan total pinalti. Hasil perbandingannya dapat dilihat pada Tabel 8.

TABEL 8. PERBANDINGAN JADWAL MANUAL VS GENETIKA

| No | Perbandingan | Jadwal manual (%) | Jadwal generate Genetika (%) | Selisih (hasil) |
|----|---|-------------------|------------------------------|-----------------|
| 1 | Tepat sasaran | 75,05 | 68,91 | -6,14 |
| 2 | Tidak tepat sasaran | 24,95 | 31,09 | -6,14 |
| 1 | Total Jadwal yang alokasi loc-nya sama persis dengan target | 75,05 | 68,91 | -6,14 |
| 2 | Total Jadwal yang alokasi loc-nya melebihi target | 5,28 | 15,45 | 10,17 |
| 3 | Total Jadwal yang alokasi loc-nya tidak mencapai target | 19,27 | 15,64 | --4,04 |
| 1 | Pinalti terbesar | 500 | 328 | (-) 172 |
| 2 | Total Pinalti | 4394 | 3695 | (-) 699 |

Tabel 8 menunjukkan bahwa dengan menggunakan algoritma genetika dapat menurunkan total pinalti dari keseluruhan jadwal sebesar 699 atau sebesar 15,9%. Penyebaran data pinalti pada kedua jadwal dapat dilihat pada Gambar 6.



Gambar 8. Perbandingan Penyebaran Pinalti LOC pada Jadwal

Gambar 8 menunjukkan bahwa penyebaran pinalti pada jadwal dengan algoritma genetika (ditandai dengan garis merah) lebih mendekati angka 0 dengan pinalti terbesar adalah 311 dan (-328). Sedangkan penyebaran pinalti pada jadwal yang masih menggunakan perhitungan manual (ditandai dengan garis biru) dengan pinalti terbesar adalah 500 dan (-411). Namun pada penyebaran pinalti tersebut terlihat bahwa jumlah alokasi LOC yang pinaltinya 0 pada jadwal dengan algoritma genetika lebih sedikit dibandingkan dengan jadwal yang manual. Dengan demikian bahwa algoritma genetika mencoba untuk melakukan proses pembuatan jadwal dengan kondisi pinalti yang seminimal mungkin dengan meminimalisir pinalti yang cukup besar.

Dari hasil uji coba yang dilakukan, dapat disimpulkan bahwa jadwal yang dibuat dengan menggunakan algoritma genetika dapat memberikan solusi optimum dibandingkan dengan jadwal yang dibuat dengan perhitungan manual. Solusi yang diberikan adalah estimasi alokasi script harian kepada PIC mendekati total LOC yang harus dikerjakan yaitu 1000 LOC.

IV. KESIMPULAN

Pada penelitian ini, kami telah melakukan pengamatan terhadap jadwal yang dibuat dengan perhitungan manual dan jadwal yang dibuat dengan menggunakan algoritma genetika. Hasil pengamatan dari uji coba pada Gambar 6 menunjukkan bahwa jadwal yang dibuat dengan menggunakan algoritma genetika dapat memberikan solusi mendekati optimal dibandingkan dengan jadwal yang dibuat dengan perhitungan manual. Hal ini ditandai dengan penurunan jumlah pinalti sebesar 15,9 %. Meskipun hasil ini tidak cukup signifikan dari hipotesa yang diharapkan tetapi algoritma genetika mencoba untuk meminimalisir pinalti yang cukup besar dengan menekan besaran pinalti agar mendekati 0.

Proses pembuatan jadwal dengan menggunakan algoritma dilakukan dengan beberapa kali percobaan sampai hasil uji coba menunjukkan bahwa solusi yang didapat memberikan hasil yang mendekati optimal. Dengan kata lain, uji coba dilakukan sampai menunjukkan tidak ada pinalti atau pinalti konvergen ke suatu nilai. Hal ini terjadi karena pada saat membentuk populasi individu yang dibentuk dengan cara pemilihan acak terhadap *script program*.

Pada penelitian yang akan datang, kami harap dapat menambahkan parameter pada saat pembuatan jadwal. Parameter tersebut adalah tingkat kemampuan *resource* dan tingkat kerumitan program dapat dipertimbangkan.

Penelitian selanjutnya dapat menggunakan metode algoritma genetika lainnya, karena pada penelitian ini masih menggunakan algoritma genetika yang sederhana.

DAFTAR PUSTAKA

- [1] D. E. Goldberg, Genetic algorithms in search, optimization, and machine learning, Reading, MA: Addison-Wesley, 1989.
- [2] K. Irfan, M. A. Rizq and A. Selly, Implementasi Algoritma Genetika Dalam Penjadwalan Perkuliahan Di Jurusan Teknik Komputer Dan Informatika Politeknik Negeri Bandung, Bandung, 2011.
- [3] L. A. Fadhil, E. C. Djamal and R. Ilyas, Optimalisasi lahan tanah untuk area rumah dan jalan menggunakan algoritma genetika, Cimahi: Universitas Wahid Hasyim Semarang, 2016.
- [4] R. Kristaningsih, H. A. Parhusip and T. Mahatma, Penggunaan Algoritma Genetika dalam Mengoptimalkan kandungan Karbohidrat dan Protein pada Mocorin, Salatiga: UNY, 2013.
- [5] M. D. Vose, The Simple Genetic Algorithm Foundation and Theory, London: Bradford Book, 1999.
- [6] S. Suyanto, Intelijensia Buatan, Bandung: STT Telkom, 2002.
- [7] Juniawati, Optimasi Proses Pengolahan Mi Jagung Instan Berdasarkan, Bogor: IPB, 2003.
- [8] R. Michael, K. Vida and S. B. Shuvr, A Modular Genetic Algorithm for Scheduling Task Graphs, United States, 2003.
- [9] P. S.N.Sivanandam and P. S.N.Deepa., Introduction to Genetic Algorithms, Berlin Heidelberg, 2008.